

Quality Report SQAS90-001

NWC Glossary of Preferred Software Engineering Terminology

October 1990

*Software Quality Assurance Subcommittee
of the
Nuclear Weapons Complex Quality Managers*

**United States Department of Energy
Albuquerque Operations Office**

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

Abstract

This glossary provides a list of terms, acronyms, and their recommended definitions. It is intended for use as an information source throughout the NWC for the preparation of documents covering software design and development, use, procurement, maintenance, and quality assurance. Use of the terms and definitions as listed in the glossary should provide for standard usage and interpretations throughout the NWC.

ACKNOWLEDGEMENT

The NWC Glossary of Preferred Software Engineering Terminology was prepared for the Department of Energy (DOE) by a Working Group of the Nuclear Weapons Complex (NWC) Quality Managers' Software Quality Assurance Subcommittee (SQAS). At the time this glossary was prepared, the Working Group had the following members:

Katherine Boissiere, DOE/AL
Cynthia Cannon, Co-editor, RF
Steve Klover, RF
Ted Nida, Chair, GEND
Lawrence Pace, DOE/PAO
Tom Putnam, PX
Melissa Smith, Y-12
Sharon Trauth, Co-editor, SNL

Use of the terms and definitions as listed in the glossary should provide for standard usage and interpretations throughout the NWC.

As an additional source for software engineering and quality assurance terminology, the Software Quality Assurance Subcommittee recommends usage of the IEEE Standards, such as ANSI/IEEE Std. 729-1983.

The Subcommittee acknowledges the help of numerous individuals and organizations in the review and development of this document. The Subcommittee also thanks Anna Donaldson of General Electric for her work in the preparation of the glossary.

SOFTWARE ENGINEERING TERMINOLOGY

Acceptance - A process which assures that software meets both performance and design requirements, is internally consistent and complete, and is ready for delivery and use.

Acceptance Equipment - A test system required for acceptance testing or inspection of weapons material, associated test and handling equipment, and joint test assemblies. Acceptance equipment may have its operations automated by software.

Adaptation Quality Factors - Characteristics governing the adaptability of the software. The adaptation factors are expandability, flexibility, interoperability, portability, and reusability.

Analysis - The examination of a topic to distinguish its component parts or elements separately or in their relation to one another.

Analyst - The organization or individual who performs the investigation, review, and design activities for the software and/or system.

Application - A specific use for which a software product is designed, based on the problem solved, the functions or operational capabilities incorporated, and on the hardware system targeted to execute the software.

Application Generator - A software development tool (itself, also software) that contains pre-defined and pre-programmed modules. Based on input supplied by the tool user, the modules can be interconnected to develop the desired application software.

As-built Design - The final design after the completion of testing and all sign offs. This final, released design should be reflected in the final software design document that becomes the basis for software maintenance.

Audit - (1) An independent assessment to check compliance with established procedures. (2) An evaluation conducted to examine the effectiveness of a process, procedure, or approach in meeting its original objectives.

Backup Copy - A copy of the software (equivalent to the master and/or working copy) that is kept in reserve for immediate use in the event of loss of the master and/or working copies. Backups must be adequately controlled, and are usually kept in magnetic format, although the hardcopy format can also be used.

Baseline - A baseline is defined as that point in the software life cycle at which formal configuration control is applied to a specific configuration.

Black-Box Testing - A software testing strategy that derives test data only from the software requirements specification. No knowledge of the internal software structure is used in developing test cases.

Boundary Value - A data value that lies on or adjacent to the boundaries of an equivalence class. [see also **equivalence class**]

Boundary Value Analysis - The process of identifying equivalence classes of input and output data, together with the boundary values for those classes. [see also **equivalence class**]

CASE - Computer-aided software engineering.

CASE Tool - A computer-aided software engineering and development tool.

Centralized Data Processing System - An interconnected set of computer hardware and software residing wholly within a single location, but used by many external locations. [contrast with **distributed data processing system**]

Certification - A process, which may be incremental, by which a contractor provides evidence to the contracting agency that a product meets contractual or otherwise specified requirements.

Change Control - A process which assures that the implementation of software changes is governed by suitable control measures commensurate with those applied to the original product, and with the scope and nature of the change. All authorized changes are processed to assure their timely incorporation at the specified effectivity points.

Classification Level - The level of security classification for the specific software.

Cohesion - (also referred to as autonomy) (1) The characteristics of the software which determine its non-dependency on interfaces and functions. (2) The degree to which tasks performed by a single program module are functionally related. [contrast with **coupling**]

Completeness - The characteristics of a piece of code or of documentation that provides full coverage of the required functions and necessary information.

Configuration - The requirements, design, and implementation, that constitute a specific software system or any of its components.

Configuration Control - The process of approving, coordinating, implementing, documenting, and verifying changes to configuration items.

Configuration Identification - The process of creating, assigning, and affixing designators to configuration items. Designators may be alphabetical, numeric, or mixtures of both. Designators are created for all facets of a configuration item, such as hardware, schematics, code, requirements, test plans, test results, and design documentation.

Configuration Item - A product, hardware or software component, subcomponent, or collection of subcomponents which is to be placed under formal configuration management.

Configuration Management - The process of identifying the configuration items in a system, controlling the release and change of those items throughout the system life cycle; recording and reporting the status of the items and associated change requests; and verifying the completeness and correctness of the items.

Context Checking - Given a software process, the action of testing the flow of information into and out of a model representing the process for coherence and consistency.

Contracted Software - Software obtained through a contractual agreement between a contracting agency and an external organization. Contracted software is custom software. [contrast with **developed software** and **procured software**]

Contracting Agency - The NWC agency, either design or production, which has prime responsibility for and authority over the software development effort.

Contractor - The organization under contract or tasking agreement with the contracting agency to perform any part of the software development effort.

Control Flow Diagram - A graphical representation of a system showing the flow of control signals throughout the system.

Controlled Conditions - An environment characterized by fully-developed and well-understood processes, documented and available work instructions, adequate equipment, trained personnel, and any required special working conditions or specifications.

Convention - Methods, formats, and patterns adopted for the presentation of information to provide consistency and understanding.

Coupling - (1) A measure of the amount of information shared between two modules. (2) A measure of the interdependence among modules in a computer program. [Contrast with **cohesion**].

CPPM - Computer Protection Program Manager.

CSCI - Computer Software Configuration Item. [see **configuration item**]

CSSO - Computer Systems Security Officer.

CTSO - Computing Telecommunications Security Organization.

Data Base - A collection or arrangement of data or data files.

Data Definition - A statement of the characteristics of basic elements of information. These characteristics may include, but are not limited to, type, range, structure, and value.

Data Dictionary - (1) A list of the names of all data elements used in a software system, together with the relevant properties of those items. (2) A set of definitions of data flows and data elements referred to in a data flow diagram.

Data Flow Diagram - A graphical representation of a system, showing the logical flow of data and the processes transforming data, together with the sinks, sources, and stores for data.

Data Sets - Information organized for analysis or used as the basis for a decision, and in a form suitable for processing by a computer.

Data Structure Diagram - A graphical representation of the ordering and accessibility relationships among items of data without regard to their actual storage configuration.

DBA - Data Base Administrator. The person(s) who provide technical training and support in data base technology and resource administration.

Deliverable - An element of the software design which is designated to be provided to the requester. Examples of potential deliverables include planning documents, draft or final design documents, test results, code, user manuals, etc.

Design Quality Factors - Characteristics governing the quality of the software design. The design factors are correctness, maintainability, and verifiability.

Developed Software - Software designed and developed in-house by an agency within the NWC. [contrast with **contracted software** and **procured software**]

Developer - The organization or individual primarily responsible for the design and implementation activities for the software system.

Development Baseline - The initial sets of specifications, design, and other documents that collectively define the software being developed to meet the approved requirements. The development baseline reflects a particular stage of development of the software product.

Distributed Data Processing System - A network of computer hardware and software located at several geographically separated locations. Each individual location is able to meet their computing needs independently, but may share information with other locations when necessary.

Documentation - (also referred to as software documentation, computer program documentation) The comprehensive written description of software in various forms and levels of detail that clearly defines its requirements, content, composition, design, performance, testing, use, and maintenance.

DOE Acceptance - The acceptance of material by the DOE based upon the results of DOE quality assurance activities and production agency quality operations.

Drawing System - The formal system for defining Sandia-designed or controlled product and test equipment. This system can be used by all NWC agencies and includes standards for the content of engineering drawings, as well as methods for their creation, storage, duplication, and maintenance.

Efficiency - The extent to which software performs its intended functions by using a minimum of computing resources.

End User - The individual or organization who will ultimately use the final software product.

Engineering Evaluation Release (EER) - A document which defines requirements to perform a Qualification Evaluation (QE), lists all planned QE activities, personnel, required data, scheduling information, data analyses, and product quantities (as applicable) required for conducting the specific QE, and which documents the specific results obtained during the QE. QEs can be conducted on hardware, manufacturing technology, software or firmware, acceptance equipment hardware and software, or on weapon system test programs. (These functions were previously referred to as Tool-Made-Sample (TMS), Qualification Sample (QS), Equipment Qualification (EQ), and Test Program Validation (TPV))

Enhancement - A change to an existing system to satisfy newly specified user requirements.

EPROM - Erasable Programmable Read Only Memory. A Semiconductor memory device (chip) which can be programmed with machine language, erased, and subsequently reprogrammed.

Equivalence Class - A class of data that is processed, or has been processed, using an identical set of rules and procedures. [see also **boundary value** and **boundary value analysis**]

Estimation - An appraisal or evaluation of the amount of time and resources required to develop software.

Evaluation - The process of determining whether an item or activity meets specified criteria.

Executable Code - A computer program (code) which is ready to be loaded into the computer. All necessary compiling, assembling, linking, etc. has been completed, and the binary code is ready for execution. [contrast with **source program**]

Expandability - The capacity to increase software capabilities or performance by enhancing current functions or by adding new functions or data.

Failure - The inability of a software system to perform a required function.

Fault - (also referred to as bug) An unintentional condition that causes software system to be unable to perform a required function (i.e., causes a failure).

Firmware - (also referred to as embedded code) Software loaded in a class of memory that cannot be dynamically modified by the computer during processing. Examples of firmware are the binary code stored in ROMs and PROMS.

Flexibility - The ease with which modifications can be made to an operational program.

Formal Test - A test conducted in accordance with approved test plans and procedures to show that the software satisfies specified requirements.

Functional Configuration Audit (FCA) - Is an audit conducted at the completion of system testing. This audit addresses the function of the software system and assures that all required fixes have been or are being implemented. It also assures that the testing was properly conducted and that the test results indicate that the system meets functional and product quality requirements. [contrast with **physical configuration audit**]

Gray-Box Testing - A testing strategy that derives test data by combining elements of black-box and white-box testing.

Hardcopy - A copy on paper or other permanent media of computer code, documentation, i.e., drawings, printed reports, plots, graphics, listings, or summaries.

Hardware - Physical equipment used in data processing, such as a computer, storage media, and other peripherals.

Header Information - That commentary information included at the beginning of a software module code listing which provides necessary identification of the module. Header information may include module name, author, variable declarations, functional description, modules called, or other appropriated information.

High Order Language (HOL) - (also referred to as higher level language) A programming language characterized by the use of nested expressions, data types, and parameter passing, in which a single statement normally generates several machine language statements. Examples of high order languages are BASIC, PASCAL, FORTRAN, COBOL, RPG II.

Historical Copy - (also referred to as revision history) A copy of software code listing or other documentation retained in retrievable form which provides a permanent record of the evolution of the software product.

IEEE - The Institute of Electrical and Electronic Engineers, Inc.

Independent Software Test Engineer - An individual, who may be a member of the software design organization, who is responsible for developing and implementing the Software Test Plan. This individual is independent in the sense that he or she is not involved in software design or code implementation.

Information System - An aggregate of software and hardware designated for delivery to a user, including the hardware, code, and procedures necessary to run the code, and all of the documentation required to fully utilize the system. When the user will perform any maintenance of the system, appropriate information is also provided.

Inspection - A formal, well-defined, peer review technique in which software is examined in detail by a group to identify defects, violations of development standards, and other problems. Inspections may be conducted on requirements, design documentation, plans, code, test results, and so forth. [contrast with **walk-through**]

Integration Testing - A structured progression of tests of the software, hardware, and combinations, which is conducted during interconnection of the software and hardware into the final system.

Integrated Tool - A software tool that combines the capabilities of two or more software development or maintenance functions in a single package. [see **CASE Tool**]

Integrity - The extent to which access to software or data by unauthorized people can be controlled.

Issue - A letter used after the drawing control number to indicate a physically different document.

Life Cycle - A period of time beginning with software conceptualization and ending with software retirement.

Life Cycle Phase - A portion of the software life cycle associated with a particular activity, such as planning, design, implementation, installation, operation, maintenance, and retirement. Each phase of the life cycle is comprised of a sequence of steps required to complete the activity of the particular phase.

Lines of Code - The number of instructions at the source code level of a programming language, excluding comment lines, that comprise a software product or module.

Local Area Network (LAN) - An interconnection of computers via telecommunications devices in one or many locations.

Maintainability - The ease with which modifications to repair or enhance the software can be implemented.

Maintenance - Modification of the software after, delivery to the user, to repair faults, improve performance, or to provide new features.

Master Copy - A copy of the software that serves as the source for all working and backup copies that are created. The master may exist on storage media or as a computer-generated or manually-written hardcopy.

Metalanguage - A language, usually with a rigid and formal syntax, in which another language may be defined.

Methodology - A body of methods, rules, techniques, and postulates employed in a field of study. [see technique]

Metrics - [see software metric]

Milestone - A scheduled event in a project used to measure progress.

Mini-Specs - [see process specification]

Model - The representation of a real-world process, device, or concept.

Module - An independently compilable software component. Modules are comprised of one or more components (procedures or routines).

Modularity - The extent to which software contains discrete components or elements such that changes to one element have minimal impact on other elements.

Non-Deliverable Software - Software that is not required to be delivered to any user.

Non-Functional Revision - A software change which is minor, such as spelling, punctuation, capitalization, and data entry or input format.

Non-Product Related Change - A change that does not affect product functionality, acceptance, or processing.

Object Program - A compiled or assembled version of the source code that is ready to be loaded into a computer.

On-Line System - A computer system (with backup capabilities in case of system failure) with a real-time capability to record, store, control, retrieve, and reproduce computer programs, information, or data.

Operating System (OS) - Software that controls the execution of a computer program and may provide scheduling, debugging, input-output control, accounting, compilation, storage assignment, data management, and related services.

Pentagon D - A designator which indicates that an item is subject to approval by the Design Agency.

Performance Quality Factors - Characteristics governing how well the software functions. The performance factors are efficiency, integrity, reliability, survivability, and usability.

Physical Configuration Audit (PCA) - Is an audit performed at the completion of the acceptance testing of the system. The primary intent of the audit is to verify that the system is ready for operation. [see functional configuration audit]

Portability - (also referred to as application independence) The ease with which software may be transported from one environment or computer system to another.

Practice - Agreed-upon methods or techniques for developing software to ensure uniformity.

Prefix - A two or three-letter identifier that designates document type.

Primitive Process - A process that has not been broken down or further divided into more processes.

Process - A unique, finite course of steps that leads to a particular result.

Process Description - [see **process specification**]

Process Specification - An unambiguous and concise description or definition of a primitive process.

Procured Software - Software obtained through purchase or lease of commercially available products, including public domain software. [contrast with **contracted software** and **developed software**]

Product Related Change - A change that affects product functionality, acceptance or processing.

Program - A sequence of instructions suitable to be executed by a computer. A program refers to the totality of software in a system or one independent part of the system software.

Program Stop - A program stop is defined as any unspecified termination of program execution which requires that the software, or a portion thereof, be reloaded, restarted, or reinitialized.

Programmer - An individual or organization responsible for providing software that conforms to the requester's requirements.

Programming Language - An artificial language designed to generate or express computer programs.

Project - An undertaking involving research, development, procurement, modification or implementation of improvements to resources or systems.

Project Folder - A file (manual or online) in which project documentation is stored.

Project Manager - The individual who coordinates and administers the project in regards to funding, project tracking, user training, and interaction; provides the project leadership for the team leaders; and is responsible to the requester for the success or failure of the project.

Project Management - The administration of a project including organization, supervision, planning, scheduling, and implementation.

PROM - (Programmable Read Only Memory) Memory that can be programmed only once after its manufacture, and only read thereafter. It is used for permanent storage of executable firmware.

Quality Factors - The general categories that inherently characterize the nature of a software project's quality. Quality factors can be grouped according to performance, design, or adaptation concerns. [see also **performance quality factors**, **design quality factors**, and **adaptation quality factors**]

Qualification Evaluation (QE) - A defined, planned series of assessment activities performed on hardware, technologies, software or firmware, acceptance equipment hardware and software, and weapon test programs. The QE encompasses evaluations previously referred to as TMS, QS, EQ, and TPV. [see also **engineering evaluation release (EER)** and **qualification evaluation release (QER)**]

Qualification Evaluation Release (QER) - A document used to summarize the results of a qualification evaluation (QE), to assign a QE status to the item, and to authorize use of the item for a specific application.

Qualification Evaluation Team (QE Team) - A group of persons responsible for implementing Qualification Evaluations.

Quality Evidence - (also referred to as quality records) Information which indicates the extent of conformance to requirements and control over processes. This information is based on process control data, inspection and acceptance activities, or traceability records.

Quality Improvement Process - An iterative process employing systematic planning, analysis of data, implementation of improvement actions, evaluation of improvement actions for effectiveness, and recognition of accomplishments.

Readability - The ease with which software can be read and understood.

Real-Time - Pertaining to the processing of data by a computer in connection with another process outside of the computer according to time requirements imposed by the outside process. Examples of real-time software include test equipment and weapon component software or firmware.

Reconfigurability - The characteristics of the computing system which provide for continuity of system operation when one or more processors, storage units, or communication links fails.

Reduced Capability Mode - After experiencing the loss of one or more system components, operation continues in a configuration which compensates for the losses but at a level below designed system capability.

Regression Testing - Testing to verify that a modified software system or unit continues to meet its specified requirements without performance degradation or failure resulting from the implementation of the modification(s).

Reliability - The probability of failure-free operation of the software for a specified time, in a specified environment.

Requirement - A condition or capability that must be met by a system or system component to satisfy a contract, specification, or other formally imposed document. The set of all requirements forms the basis for system development.

Reusability - The extent to which the software, or any of its modules, can be used in other applications.

Review - A technical examination of a product or one of its components for the purpose of detection and remedy of deficiencies.

Robustness - (also referred to as anomaly management) (1) The characteristics of the software which provide for continuity of operations under and recovery from adverse conditions. (2) The extent to which software can continue to operate despite the introduction of invalid inputs.

ROM - (Read Only Memory) A memory device that can only be read after it is manufactured. It is used for permanent storage of firmware.

Software - Computer programs, procedures, rules, and associated documentation and data pertaining to the operation of a computer system.

Software Change - All changes to software (documentation and/or code) that result in software being different from the original release or previously changed copy.

Software Change Proposal - A change, which has not yet been implemented, to existing code, its documentation, or its interfaces which would alter the approved baseline software which is under software configuration control.

Software Control Identifier - A unique alpha/numeric designator assigned to provide identification and traceability for software. This includes reference to change indicators such as version, revision, or suffix.

Software Design - The definition of software architecture, components, modules, interfaces, testing approach, and necessary test and operational data for software to satisfy specified requirements.

Software Development - The engineering process that results in software, encompassing the span of time from initiation of the contracted effort through delivery to and acceptance by the requester.

Software Development Project - The overall task of producing software. This may encompass internal design, external procurement, or both.

Software Development Library - A controlled collection of software, documentation, and associated tools and procedures used to facilitate the orderly development and subsequent support of software. A software development library provides storage of and controlled access to software and documentation in both human-readable and machine-readable form. The library may also contain management data pertinent to the software development project.

Software Engineering - The systematic approach to the specification, design, development, testing, operation, maintenance, and retirement of software.

Software Hierarchy - The breakdown of software into components/elements. Examples include modules, units, programs and subprograms.

Software Libraries - Controlled access storage areas that contain software code in development, test, or production mode.

Software Library Custodian - An individual designated by management to control access to master copies of software programs.

Software Maintenance Methodology - That body of methods, rules, and procedures employed to accomplish the individual steps needed to properly manage maintenance.

Software Manager - Individual/Organizations responsible for software/hardware maintenance and for configuration control of an installed programmable system.

Software Media - The medium(s) upon which the software resides. Software medium(s) include punched tape, magnetic tape, magnetic fixed disk, floppy disks, PROMS, EPROMS, etc.

Software Metric - A qualitative measure of the processes used to produce software, or of a specific software product.

Software Plans - A collective term used to describe the procedures and standards for software project management, software engineering, software test, software qualification evaluation, software configuration management, and software quality assurance.

Software Problem Report - A record of a software deficiency, such as a failure or an error in documentation.

Software Quality - (1) The totality of features and characteristics of a software product that bear on its ability to satisfy the requirements specification. (2) The degree to which the software conforms to customer requirements and expectations.

Software Quality Assurance - A planned and systematic pattern of all actions necessary to provide adequate confidence that a software product conforms to established technical requirements.

Software Quality Assurance Plan (SQAP) - A plan written at the software project level which states specific software quality assurance activities to be implemented on the project.

Software Quality Control - Those quality assurance actions that provide a means to control and measure the characteristics of software to established requirements.

Source Program - A computer program that must be compiled, assembled, or otherwise interpreted prior to execution by the computer.

SRS - Software requirements specification

Stand-Alone Tool - A software tool that provides a single development or maintenance capability in one system.

Standard - Procedures and rules employed and enforced to prescribe a disciplined, uniform approach to software development.

States - The conditions or modes of behavior of a system.

State Transition Diagram - A graphical representation of the states of a system and the way transitions from one state to another occur.

Structure Chart - A graphic chart for depicting the partitioning of a software system into modules, the hierarchy and organization of these modules, and the communication interfaces between the modules.

Structured Analysis - A disciplined approach to requirements specification that stresses the development of a maintainable specification and communication with the user. The analysis is based on a technique using control flow diagrams, data flow diagram, a data dictionary, and process specifications, as applicable, to define system inputs, processes, and outputs.

Structured Design - A disciplined approach to the design of software based on a hierarchical decomposition and implementation of system functions into software modules. Structured design has as one of its goals to maximize cohesiveness and minimize coupling.

Structured Methods - Software development methods that are concerned with components of a system and the interrelationship between these components. Structured methods include structured analysis, structured design, and structured programming.

Structured Programming - A well-defined software development technique that normally incorporates strict use and implementation of three basic control constructs: sequence; selection; and repetition.

Subcomponent - (1) Lower level design components; (2) The components of the components.

Subcontractor- A manufacturer or supplier to a production agency, contractor, or to another subcontractor, who will provide major components or special designed items for the DOE.

Subprogram - A major functional subset of a program made up of one or more modules.

Support Software - A computer program, and associated documentation, used in general support of the processes of a computer (such as diagnostic, copy, trace, or sort programs), or to assist in software design (such as CASE Tools, test software, development systems, and compilers).

Synchronization - A process of maintaining one or more operations in step with each other.

System - An Integrated whole that is composed of diverse, interacting specialized structures, performing many duties by functioning as a single entity.

System Manager - The individual or organization or group who provides technical support for the management and operation of computer systems.

System Software - Software that controls the execution of computer programs and may provide scheduling, debugging, input-output, accounting, compilation, storage assignment and data management. [see **operating system**]

Technique - A systematic means of achieving a desired goal. [contrast with **methodology**]

Test Case - A specific test definition developed for a particular objective or to verify a particular characteristic of the software. Test cases include test data, objectives, and expected outcomes.

Testing - The process of exercising a system (or some component) to identify differences between expected and actual results.

Test Library - A controlled collection of software designed to aid in testing other software.

Test Plan - A written document that outlines the methods and philosophy to be followed in testing the software.

Test Procedure - Documentation specifying a sequence of actions necessary to conduct a particular test, using a specific set of test cases. A test procedure may apply to more than one test case.

Test Team - A group of persons responsible for the development, execution, and proper documentation of the Test Plan.

Top-Down - Pertaining to an approach that starts with the highest level of a hierarchy and proceeds through progressively lower levels. For example, top-down design, top-down coding, top-down testing.

Traceability - The extent to which software provides an audit trail from the requirements to the implementation, and between subsequent revisions.

Training - A process of instructing or familiarizing an individual in areas of software engineering, software operation, etc.

Unit - The smallest logical entity specified in the detailed design which completely describes a single function in sufficient detail to allow code implementation and testing independently of other units. Units are the actual physical entities implemented in code.

Use Control - (also referred to as access control) A method or system to ensure that only approved versions of controlled software are utilized by authorized individuals.

User - An individual or organization deriving primary benefit from the programmed system.

User-Interface - The means by which a user interacts with or communicates with a computer system. The interface involves both visual and behavioral actions and reactions of the user.

User-Interface Prototyping - A prototyping technique that focuses on the development of the interface between the user and the computer.

Validation - The evaluation of software, at the end of the development process to assure that specified requirements are met. [see also **verification**]

Verifiability - The effort required to test and verify that the software performs its intended function.

Verification - The process of determining whether or not products of a given phase in the software development cycle fulfill the requirements established during the previous phase. [see also **validation**]

Walk-through - An informal review technique (usually less structured than inspections) in which the designer (or author) guides participants through an examination of software for problems. Walk-throughs may be conducted on requirements, design documentation, plans, code, test results, and so forth. [contrast with **inspection**]

Waterfall Model - A model of the software life cycle in which the time periods or phases of the life cycle are arranged as the cascades in a waterfall. The software life cycle typically includes a concept exploration phase, a requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and retirement phase.

Weapon Material - DOE nuclear weapons, assemblies, components, or parts thereof, and associated test and handling equipment.

Weapon-Related Material - Any material other than weapons material being developed and produced for or by the DOE and intended for use in conjunction with or in any way related to weapons.

White-Box Testing - A software testing strategy that derives test data from knowledge of the program's internal structure.

Working Copy - An officially released, unaltered copy of the master software program stored on the appropriate computer medium, usually maintained at the operating location, and used to run the system.